

**REMARKS**

At the time the current Final Official Action was mailed, the Examiner rejected claims 1-21 and 23-35. In view of the following remarks, Applicant respectfully requests reconsideration and allowance of all pending claims.

**Rejections under 35. U.S.C. § 112**

The Examiner rejected claims 1, 15, and 21 under 35 U.S.C. § 112 as being indefinite because the claims do not clearly redefine the term “a call of a command line utility...wherein the command line utility is a utility executable from a command line prompt.” Specifically, the Examiner stated:

Where applicant acts as his or her own lexicographer to specifically define a term of a claim contrary to its ordinary meaning, the written description must clearly redefine the claim term and set forth the uncommon definition so as to pct one reasonably skilled in the art on notice that the applicant intended to so redefine that claim term. *Process Control Corp. v. HvdReclaim Corp.*, 190 F3d 1350,1357,52 USPQ2d 1029, 1033 (Fed. Cu. 1999). The term “a call of a command line utility...wherein the command line utility is a utility executable from a command line prompt” in claims 1, 15, and 21 is used by the claim to mean “a utility that is capable of being executed from a command line prompt” (prompt meaning some type of visual indicator, such as a flashing’>’ character in a shell terminal interface or user interface), while the accepted meaning is literally “a utility that is executed from a command line prompt”. The term is indefinite because the specification does not clearly redefine the term.

...

As defined in *Operating Systems Second Edition*, by 1-1. M. Deitel (1990). p. 574: A command line consists of a command name (i.e. the name of an executable file), followed by a list of arguments separated by blanks. Examiner requests a clarification, and indication of support shown in the Specification, that an application invokes a call, of a command line utility, wherein the application call provides an identifier, within a shell terminal or user interface, as Applicant’s amendment is suggesting.

Office Action, pages 5-6.

Applicant respectfully traverses this rejection.

Applicant is *not* acting as his own lexicographer with regard to the claim term “a call of a command line utility...wherein the command line utility is a utility executable from a command line prompt.” Therefore, the claim term should be given its plain meaning, i.e. the “ordinary and customary meaning given to the term by those of ordinary skill in the art.” M.P.E.P. 2111.01. The ordinary and customary meaning of a term may be evidenced by a variety of sources, including “the words of the claims themselves, the remainder of the specification, the prosecution history, and extrinsic evidence concerning relevant scientific principles, the meaning of technical terms, and the state of the art.” *Phillips v. AWH Corp.*, 415 F.3d 1303, 1314, 75 U.S.P.Q.2d 1865, 1876 (Fed.Cir. 2005). If extrinsic reference sources, such as dictionaries, evidence more than one definition for the term, the intrinsic record must be consulted to identify which of the different possible definitions is most consistent with applicant's use of the terms. *Brookhill-Wilk I, LLC v. Intuitive Surgical, Inc.*, 334 F.3d 1294, 1300, 67 U.S.P.Q.2d 1132, 1137 (Fed.Cir.2003).

The term “command line” is defined as “the line on the screen that has the prompt; the line where the next command that is typed will appear.” Dictionary of Computer and Internet Words: An A to Z Guide to Hardware, Software, and Cyberspace, 53 (1st ed. 2001). Another definition of “command line” is “in a shell, a command typed by a user.” Dictionary of Computer Science, Engineering, and Technology (Phillip A. Laplante ed., 2001). Yet another definition of “command line” is “an area where commands are typed in a command line user interface.” Bryan Pfaffenberger, Webster’s New World Computer Dictionary 80 (10th ed. 2003). All of these definitions are consistent with the specification and claims of the present application.

According to the plain meaning as evidenced by the dictionary definitions above, a “command line” is a line consisting of a prompt and/or any commands, and a “command line prompt” is a shell or prompt where commands are typed. A “command line utility,” as stated in the specification and claims, is a “utility executable from the command line prompt.” For

example, as is known to a person having ordinary skill in the art, the “dir” command discussed in the specification is a command that must be typed into a “command prompt” in order for execution by the operating system. Application, page 4.

In the WINDOWS operating system, such a “command prompt” is also referred to as an “MS-DOS” box or window. The Microsoft Computer Dictionary cited by the Examiner defines “dir” as “an MS-DOS command...” Microsoft Computer Dictionary 161 (5th ed. 2005). Further, a “command prompt window” is defined as “a window displayed on the desktop used to interface with the MS-DOS operating system...commands are typed at an entry point identified by a blinking cursor.” *Id.* at 111-112. Therefore, in the context of a WINDOWS embodiment, a command line utility, e.g. the “dir” command, is typed into a command prompt, e.g. MS-DOS window. Thus, the specification is not trying to redefine the term “a call of a command line utility...wherein the command line utility is a utility executable from a command line prompt” as suggested by the Examiner, but is simply relying on the plain meaning of the term. Therefore, Applicant respectfully requests withdrawal of the §112 rejection.

### **Rejections under 35 U.S.C. § 103**

The Examiner rejected claims 1-21 and 23-25 under 35 U.S.C. § 103(a) as being unpatentable over Buxton (U.S. Patent No. 6,182,279, hereinafter “Buxton”) in view of Qureshi (U.S. Patent No. 5,758,154, hereinafter “Qureshi”). Applicant respectfully traverses the rejection.

The burden of establishing a *prima facie* case of obviousness falls on the Examiner. *Ex parte Wolters and Kuypers*, 214 U.S.P.Q. 735 (PTO Bd. App. 1979). Obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention absent some teaching or suggestion supporting the combination. *ACS Hospital Systems, Inc. v. Montefiore Hospital*, 732 F.2d 1572, 1577, 221 U.S.P.Q. 929, 933 (Fed. Cir. 1984). Accordingly, to establish a *prima facie* case, the Examiner must not only show that the

combination includes *all* of the claimed elements, but also a convincing line of reason as to why one of ordinary skill in the art would have found the claimed invention to have been obvious in light of the teachings of the references. *Ex parte Clapp*, 227 U.S.P.Q. 972 (B.P.A.I. 1985).

As preliminary matter, Applicant asserts that the Examiner's taking of Official Notice that "an executable command to modify the registry, is capable of being executed from a command line prompt" is inappropriate and unsupported. As stated in M.P.E.P. 2144.03, "[o]fficial notice without documentary evidence to support an examiner's conclusion is permissible only in some circumstances. While 'official notice' may be relied on, these circumstances should be rare when an application is under final rejection or action under 37 C.F.R. 1.113. Official notice unsupported by documentary evidence should only be taken by the examiner where the facts asserted to be well-known, or to be common knowledge in the art are capable of instant and unquestionable demonstration as being well-known." M.P.E.P. 2144.03. Thus, where there is a final rejection, and Examiner has provided no documentary evidence that such a command to modify the registry is executable from the command prompt, the Official Notice is inappropriate at this time.

Additionally, Applicant would like to point out that the Microsoft Computer Dictionary used by the Examiner is dated 2005 and is published by Microsoft, manufacturer of the WINDOWS operating system and many WINDOWS software products. As such, this dictionary may not provide the definitions consistent with claim terms that apply to multiple operating systems. The Microsoft Computer Dictionary may contain definitions or interpretations specific to Microsoft or WINDOWS, and may not be the best indicator of the plain meaning of a term when the term is clearly applicable to multiple operating systems (such as both WINDOWS and Unix operating systems). Further, although the embodiment discussed in the specification and discussed in the Examiner's rejections is a WINDOWS operating system having a system registry, the claims and claim terms should not be limited to such an embodiment.

***First §103 reference (Buxton)***

In characterizing Buxton, the Examiner stated:

Buxton inherently invokes a utility (system level services/ registry editor, col. 8, line 7) to modify and store the customized components created. An identifier is inherently provided to register the customized component in the registry. Col. 7, line 65-col. 8 line 10, “Container may comprise any stand alone application capable of embedding OLE controls. A container **interacts with the WIN32 APIs** through the OLE libraries **in order to insert** OLE objects or controls **into the operating system registry**...The OLE libraries function to call the WIN32 APIs to locate (using a type of identifier) registered objects in registry and to insert and create object dialog (utility calls identify objects inserted / created / modified in the registry)”. (emphasis added)

Office Action page 7.

Applicant respectfully disagrees with the Examiner’s characterization of the Buxton reference. The Examiner ignored an important distinction between a *utility* and an *API* (application programming interface). As defined in the Microsoft Computer Dictionary cited by the Examiner, a utility is “a *program* designed to perform a particular function.” (Emphasis added). Microsoft Computer Dictionary 544 (5th ed. 2005). In contrast, an application programming interface is defined as “a set of routines used by an application program to direct the performance of procedures by the computer’s operating system.” *Id.* at 33. Finally, OLE is defined as “a technology for transferring and sharing information among applications.” *Id.* at 375.

Buxton discloses an application, container 220, that uses OLE libraries and WIN32 APIs to access the registry. Buxton, col. 7, lines 65-67; col. 8, lines 1-2. As evidenced by the definitions above, OLE libraries and the WIN32 API are *not* a “command line utility” executable from a command line prompt as recited in the independent claims. For example, although Buxton discloses a WINDOWS embodiment, Buxton does not refer to an MS-DOS box or window or any command executed from such a window. As discussed above, an MS-DOS window is the “command line prompt” of a WINDOWS operating system. Thus, Buxton does

not disclose a “command line utility” executed from a “command line prompt.” Accordingly, any of the Examiner’s further assertions that rely on Buxton disclosing a “command line utility” are incorrect.

Additionally, with regard to the Buxton reference, the Examiner stated:

Buxton suggested receiving commands via command line, which results in modifying the registry (system storage) and storage. Buxton failed to specifically disclose a “command line utility”. Buxton suggests that the command line input (an object that consists of modifications to base component) is directed (DIR utility — a command utility) to storage, and the registry is edited (a command utility), but did not explicitly disclose ‘command line utility’.

Office Action, page 8.

Applicant asserts that Buxton does not disclose “receiving output from the command line utility” or “storing the command line utility output...” as recited in the independent claims, nor are the Examiner’s statements accurate. The Examiner stated that “directing to storage” is accomplished through the DIR command line utility, and the registry is edited by a command line utility. This statement is not a correct definition of the DIR command line utility. The “DIR” utility does not “direct to storage” as the Examiner suggested, but instead is a directory command in Microsoft WINDOWS that lists the contents of a directory in the file system. The DIR command, while a command line utility in WINDOWS, does not “direct to storage” and therefore Buxton cannot be read as disclosing this command (nor any other command line utility). As a consequence, Buxton is incapable of disclosing “receiving output” from a command line utility or “storing the command line utility output.”

Finally, editing the registry, in and of itself, does not disclose the claim element of “storing the command line utility output in a system storage...” as recited in the independent claims. Accessing, editing, and modifying the registry in a Microsoft WINDOWS operating

system is typically performed through OLE libraries and WIN32 APIs. As stated above, neither technique is a “command line utility” as disclosed in the present Application.

*Second §103 reference (Qureshi)*

With regard to Qureshi, the Examiner stated:

Qureshi clearly discloses (col. 3, lines 26-46) an application call b a registration routine (‘command line utility’) via system calls, to store an identifier in the registry. Col. 4, lines 2-7, “When a computer program invokes (commands) the registration routine, it passes to the registration routine a configuration file that contain a description of the configuration information of the computer program. The registration routine open the configuration file and add the configuration information to the registry.” Similar to Buxton’s invention, Qureshi disclosed (col. 4, lines 58-67, ‘Moreover, when a new component is added to an already installed application, the user previously was required to re-execute the entire installation procedure. With the registration DllRegisterServer routine, a new SRG tile that includes only the configuration information required by the new component can be distributed along with the new component, and the **registry can be properly updated by simply calling DllRegisterServer (a command line utility call) and supplying as an argument the pathname of the SRG file containing the configuration information required by the new component.**” (emphasis added) Col. 5, lines 56-59, “The application programs invoke operating system routines (invoking by an application, a call of a command line utility) in order to access the various services provided by the operating system, including routines for reading and writing files. **The application programs invoke the registration. ...routines** provided by the registration OLL to register.. configuration information, passing an indication SRG file that contains the configuration information to register (**identifier**). **..The registration...routines** of the registration DLL **call operating system routines (utilities) to write information to the registry file...**” (emphasis added) Col. 8, lines 4847, “DllRegisterServer opens the specified key.. .calling the system routine RegOpenKey.. .DllRegisterServer creates the key in the registry. . . DllRegisterServer then calls the subroutine Process\_key\_values, . . The subroutine.. .returns TRUE. . .and returns FALSE...”

Office Action, page 8-9.

Again, Applicant disagrees with the Examiner's characterization of the prior art reference. The DllRegisterServer *routine* disclosed in Qureshi is *not* a command line utility. In fact, the Qureshi reference never mentions the words "command line," "command line prompt," or "utility." DllRegisterServer is a *routine* that is a part of a DLL (dynamic link library) that is not intended to be or capable of being called from a command line. A routine is defined as "any section of code that can be invoked within a program." Microsoft Computer Dictionary 161 (5th 2005). A routine is *not* a command line utility that may be invoked from a command line; by its very nature, a *routine* is invoked from within a program. It is impossible for the DllRegisterServer *routine* to be called from the command line, and therefore the DllRegisterServer disclosed in Qureshi cannot be a "command line utility" as recited in the independent claims. Thus, the Examiner's use of Qureshi to disclose a "command line utility" is incorrect.

Accordingly, neither Buxton nor Qureshi, cited alone or in combination do not disclose or suggest all of the elements of the claimed invention, and thus, cannot possibly render the claimed subject matter obvious. Therefore, Applicant respectfully requests withdrawal of the Examiner's rejection and allowance of independent claims 1, 15 and 21 and the claims that depend therefrom.

### **Conclusion**

In view of the remarks and amendments set forth above, Applicant respectfully requests reconsideration of the Examiner's rejections and allowance of all pending claims. If the Examiner believes that a telephonic interview will help speed this application toward issuance, the Examiner is invited to contact the undersigned at the telephone number listed below.




**General Authorization for Extensions of Time and Payment of Fees**

In accordance with 37 C.F.R. § 1.136, Applicant hereby provides a general authorization to treat this and any future reply requiring an extension of time as incorporating a request therefor. Furthermore, Applicant authorizes the Commissioner to charge the appropriate fee for any extension of time to Deposit Account No. 06-1315; Order No. MICS:0194/FLE/MAN. Further, the Commissioner is authorized to charge any other fees that may be due at this time or at time during the pendency of this application to the Deposit Account listed.

Respectfully submitted,

Date: May 23, 2007

  
Robert A. Manware  
Reg. No. 48,758  
FLETCHER YODER  
P.O. Box 692289  
Houston, TX 77269-2289  
(281-970-4545)